# An Anonymous Inter-Network Routing Protocol for the Internet of Things

Paolo Palmieri[1], Luca Calderoni[2] and Dario Maio[2]

[1]*Cranfield University, Centre for Electronic Warfare Information and Cyber Shrivenham, Swindon SN6 8LA, United Kingdom*
[2]*University of Bologna, Dept. of Computer Science and Engineering Cesena, 47521, Italy*
*Email: paolo.palmieri@cranfield.ac.uk; {luca.calderoni, dario.maio}@unibo.it*

Received 16 April 2017; Accepted 27 September 2017;
Publication 31 October 2017

## Abstract

With the diffusion of the Internet of Things (IoT), computing is becoming increasingly pervasive, and different heterogeneous networks are integrated into larger systems. However, as different networks managed by different parties and with different security requirements are interconnected, security becomes a primary concern. IoT nodes, in particular, are often deployed "in the open", where an attacker can gain physical access to the device. As nodes can be deployed in unsurveilled or even hostile settings, it is crucial to avoid escalation from successful attacks on a single node to the whole network, and from there to other connected networks. It is therefore necessary to secure the communication within IoT networks, and in particular, maintain context information private, including the network topology and the location and identity of the nodes.

In this paper, we propose a protocol achieving anonymous routing between different interconnected networks, designed for the Internet of Things and based on the spatial Bloom filter (SBF) data structure. The protocol enables private communication between the nodes through the use of anonymous identifiers, which hide their location and identity within the network. As routing information is encrypted using a homomorphic encryption

*Journal of Cyber Security, Vol. 6_2,* 127–146.
doi: 10.13052/jcsm2245-1439.622

scheme, and computed only in the encrypted domain, the proposed routing strategy preserves context privacy, preventing adversaries from learning the network structure and topology. This, in turn, significantly reduces their ability to gain valuable network information from a successful attacks on a single node of the network, and reduces the potential for attack escalation.

**Keywords:** Internet of Things, Privacy-preserving Technologies, Anonymous Routing, Spatial Bloom Filters.

## 1 Introduction

The devices and networks that compose the Internet of Things (IoT) are extremely diverse and heterogeneous in terms of resources, capabilities, lifespan and communication technologies. New IoT products are constantly being introduced to the market, and each device is able to gather (for example through sensors) and process increasing amounts of information. Billions of smart objects are being immersed in the environment, sensing, interacting, and cooperating with each other to enable cities and services to become smarter, with great benefits to the environment, the economy and the society as a whole. However, if security and privacy concerns are not addressed, the Internet of Things could also present opportunities for malicious attackers, exploiting the vulnerabilities of devices that are not always designed with security in mind. In fact, attacks against smart devices, sensor networks and the Internet of Things are increasing in both frequency and magnitude [12].

The emerging security threat, however, is not slowing the growing diffusion of systems and services based on IoT and heterogeneous sensor networks, propelled by the relentless advances in the production of low-cost embedded devices and sensors. As these technologies are usually deployed in wireless environments, Wireless Sensor Networks (WSN) have become a suitable solution for an increasing number of applications, including health monitoring, smart agriculture, weather sensing, intrusion detection applications and industrial control [8, 13]. In urban and suburban contexts, these networks are often connected to each other, and ultimately to the Internet, enabling remote monitoring and management as well as autonomous computation. In spite of extensive research in the area, the Internet of Things and in particular the interconnection of IoT devices and networks still pose a significant security and privacy risk [11].

IoT devices are often deployed in unsecured areas or outdoor, where they can be subject to tampering, leading to a potential attacker being able to gain

control of one or more nodes in the IoT network. The use of wireless communication also makes the network inherently vulnerable to eavesdropping. For these reasons, it is crucial to design and embed in IoT networks security mechanisms and protocols than can preserve the security of the network even in the presence of internal adversaries. In particular, both the communication between the nodes participating in the network and the context information (including the location of the nodes and the network organization) should be protected [11]. In order to preserve the former, and guarantee the privacy of the communication, nodes can employ encrypted communication protocols. Research in this area is being targeted at the design of efficient cryptographic systems, due to the low-power nature and limited computational capabilities of most IoT devices. In this paper we focus instead on the latter challenge: preserving context privacy, that is, protecting information on the network topology, structure and organization, including the identity and network location of the nodes in the network [4]. As nodes in IoT network have different roles, context privacy is crucial to prevent targeted attacks on important nodes, such as the ones used to aggregate information: an example are WSNs, which are in general highly vulnerable to attacks targeted at base stations (the nodes collecting the data gathered by the sensors). Failure of a base station can disrupt operation of the entire network, making it an ideal target for an attacker. In order to prevent adversaries from launching both remote, software-based attacks and physical attacks, the location of base stations and the network topology should therefore be concealed [5].

### 1.1 Literature Review

A basic strategy to achieve context privacy is to use flooding and transmissions of fake or dummy packets, which make network traffic observation more difficult [21]. However, this solution introduces significant overheads in the communication, and can reduce the efficiency of the IoT network. More complex strategies are normally based on some flavor of anonymity, including the use of random walks to route packets anonymously [10]. Random walks have been adopted in a number of designs: Zhang proposed self-adjusting directed random walks in [22], while GROW (Greedy Random Walk) [20] introduced a two-way random walk, from both source and destination, that can reduce the chance of an eavesdropper being able to collect location information. Finally, layers of encryption can be used to protect the information at each hop in the walk [6].

More recently, more advanced anonymity techniques have been applied to the IoT. Black routing and node obscuring for IoT have been proposed

by Chakrabarty et al. in [3]. Their strategy hides the source of network traffic via a token-based routing approach, while the destination is obscured by forwarding the packet beyond the final destination. However, to achieve anonymity of source-destination pairs, a minimum of 40% of the total IoT nodes in the path is needed, thus restricting application of this technique to more complex settings, where different IoT networks are interconnected. An onion routing protocol derived from Tor has also been designed for the Internet of Things scenario [15]. This strategy, however, requires IoT nodes to be able to perform complex computations, which may not always be possible in power and resource constrained scenarios.

### 1.2 Contribution

In this paper we introduce a novel anonymous routing mechanism, based on the spatial Bloom filter data structure and homomorphic encryption. Part of the results of this article were previously presented by the authors in a shorter version at the International Workshop on Malicious Software and Hardware in the Internet of Things, co-located with Computing Frontiers 2017 [17]. The proposed construction is targeted at preserving context privacy within a network composed of a number of interconnected subetworks. In particular, our construction can find direct application in all the settings where different networks, such as wireless sensor networks or networks of smart or embedded devices, are connected to form a larger network. The anonymous routing mechanism achieves the following goals: encrypt communication between nodes; hide the identity and location of the sending and receiving nodes in a communication between two different subnetworks; hide the network structure and topology to all the nodes; and hide the origin and destination of any communication between subnetworks to the routing layer (that is, the network infrastructure that connects the different subnetworks and is responsible for the routing of packets between them). These properties enable context privacy and security against adversaries who control one or more nodes within the network, and prevent attacks aimed at taking over control of the network.

## 2 Preliminaries

In the following we present the main building blocks of the proposed routing mechanism: first, the spatial Bloom filter (SBF) [2, 16]. Second, the homomorphic encryption operations that make it possible to compute the SBF in

the encrypted domain. For the latter, we base our construction on the Paillier cryptosystem [14], although any equivalent alternative cipher may be used.

### 2.1 Spatial Bloom Filters

A Bloom Filter (BF) is a data structure that represents a set of elements in a space-efficient manner [1]. Bloom filters are widely used in networking protocols, and have a variety of network security applications [9]. Recently, Calderoni, et al. proposed a compact data structure based on Bloom filters, designed to store location information [2, 16]. The structure, called spatial Bloom filter (SBF), was originally designed for location privacy applications. Two private positioning protocols were proposed with the SBF, both aimed at keeping both the user's exact position and the provider's monitored areas private. The SBF was recently evaluated in a comparative assessment with other similar privacy-preserving techniques, showing promising properties in several domains [19]. In particular, the SBF is suitable for application beyond the location privacy field. In this paper, we use the SBF to build a novel private routing protocol for interconnected networks, a typical scenario in the IoT and distributed sensor networks domain. In the following, we briefly review the data structure and its properties relevant to the proposed construction; a full discussion of the primitive can be found in [2, 16].

A spatial Bloom filter extends the original Bloom Filter idea in order to support several sets composed of elements belonging to a specific domain $\mathcal{E}$. A SBF can be used to perform membership queries on the originating set of elements without knowledge of the set itself but, contrary to the BF, a SBF can be constructed over multiple sets. Querying a spatial Bloom filter for an element returns the identifier of the specific set among all the originating sets in which the element is contained, minus a false positive probability. The false positive probability depends on a number of parameters chosen during the filter construction, which can be selected to achieve a desired probability.

Let $\mathcal{E}$ be a domain specific set of elements (in this paper elements represent the IDs of network nodes); a SBF can be defined as follows [2]:

**Definition 1.** *Given the originating sets $\Delta_1, \Delta_2, \ldots, \Delta_s$ to be represented in the filter, let $\bar{S}$ be the union set $\bar{S} = \bigcup_{\Delta_i \in S} \Delta_i$ and let $S$ be the set of sets $S = \{\Delta_1, \Delta_2, \ldots, \Delta_s\}$. Let $O$ be the strict total order over $S$ for which $\Delta_i < \Delta_j$ for $i < j$. Let also $H = \{h_1, \ldots, h_k\}$ be a set of $k$ hash functions such that each $h_i \in H : \{0,1\}^* \rightarrow \{1, \ldots, m\}$, that is, each hash function in $H$ takes binary strings as input and outputs a random number uniformly*

*chosen in* $\{1, \ldots, m\}$. *We define the* spatial Bloom filter *over* $(S, O)$ *as the set of couples*

$$B^{\#}(S, O) = \bigcup_{i \in I} \langle i, \max L_i \rangle \tag{1}$$

*where* $I$ *is the set of all values output by hash functions in* $H$ *for elements of* $\bar{S}$

$$I = \bigcup_{\delta \in \bar{S}, h \in H} h(\delta) \tag{2}$$

*and* $L_i$ *is the set of labels* $l$ *such that:*

$$L_i = \{l \mid \exists \delta \in \Delta_l, \exists h \in H : h(\delta) = i\} \quad . \tag{3}$$

A spatial Bloom filter $B^{\#}(S, O)$ can be represented as a vector $b^{\#}$ composed of $m$ values, where the $i$-th value

$$b^{\#}[i] = \begin{cases} l & \text{if } \langle i, l \rangle \in B^{\#}(S, O) \\ 0 & \text{if } \langle i, l \rangle \notin B^{\#}(S, O) \end{cases} \quad . \tag{4}$$

For reference, in Table 1 we provide the notation commonly used within the SBF domain.

The construction of an SBF starts by setting all values in $b^{\#}$ to 0. Then, starting from the first set $\Delta_1$, each element belonging to the set is

**Table 1** A list of symbols commonly used in SBF literature

| Symbol | Description |
|---|---|
| $\mathcal{E}$ | A domain for elements to be mapped inside a SBF |
| $B^{\#}(S, O)$ | A spatial Bloom filter |
| $\langle x, y \rangle$ | The pair composed by $x$ and $y$ |
| $b^{\#}$ | The vector representation of the SBF |
| $b^{\#}[i]$ | The $i$-th cell (position) of the SBF vector representation |
| $k$ | The number of hash functions |
| $m$ | The number of cells of the SBF |
| $n$ | The total number of elements to be inserted into the SBF |
| $s$ | The number of originating sets |
| $\Delta_i$ | The $i$-th originating set |
| $S$ | The set of originating sets $\{\Delta_1, \ldots, \Delta_s\}$, $\lvert S \rvert = s$ |
| $\bar{S}$ | The union set $\bigcup_{\Delta_i \in S} \Delta_i$, where $\lvert \bar{S} \rvert = n$ |
| $O$ | The strict total order over $S$ ($\Delta_i < \Delta_j, i < j$) |
| $L$ | The set of set labels $\{1, \ldots, s\}$ |

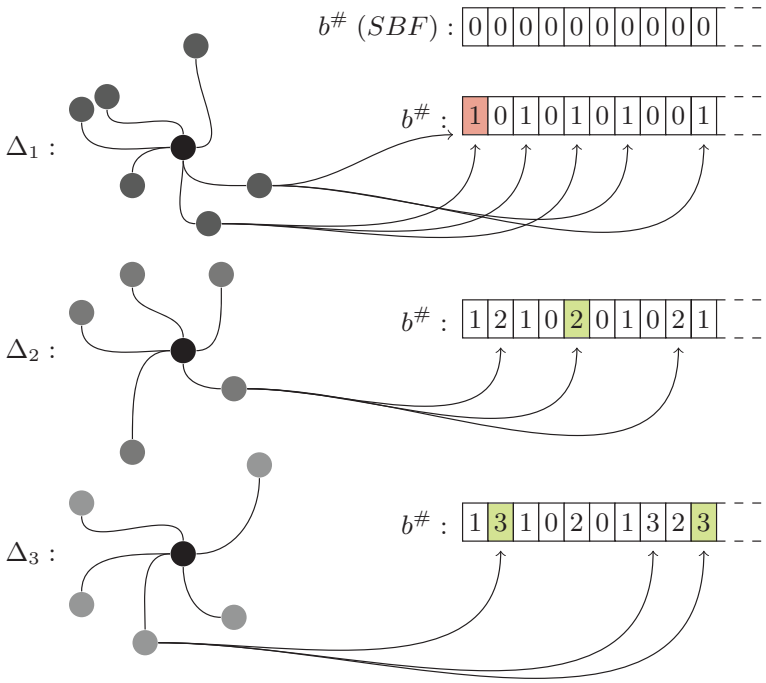


**Figure 1** Sets $\Delta_1$, $\Delta_2$ and $\Delta_3$ (representing three subnetworks) are used to construct a SBF. Three hash functions are used to map each element into the filter. In the first step of this example, the identifiers of two nodes belonging to $\Delta_1$ are processed by the hash functions, resulting in the value '1' being written six times into the SBF. The construction proceeds likewise for elements of $\Delta_2$ and $\Delta_3$. Two kinds of collisions are possible, as highlighted: the first is intra set; the second takes place when elements of sets marked with a greater label overwrite those with a lower value. The probability of both events can be controlled to prevent false positives.

processed by each function $h \in H$. Let us suppose $h(\delta) = i$: in that case, the $i$-th value of $b^{\#}$ will be set to 1 (as 1 is the label associated to $\Delta_1$). Elements belonging to subsequent sets $(\Delta_2, \ldots, \Delta_s)$ are processed likewise. It is important to note that collisions between two distinct values are subject to the SBF collision rule: labels with higher value overwrite those with lesser value. This procedure is exemplified in Figure 1.

In order to check whether or not an element $\delta_u$ is member of the set $\Delta_i \in S$, two conditions need to be met:

$$\exists h \in H : b^{\#}[h(\delta_u)] = i \quad \text{and} \quad \forall h \in H, b^{\#}[h(\delta_u)] \geq i \ . \tag{5}$$

Substantially, one single $b^{\#}\left[h(\delta_u)\right] = 0$ is sufficient to state that $\delta_u$ is not a member of $\bar{S}$. On the contrary, if $b^{\#}\left[h(\delta_u)\right] \neq 0$ for each hash function, then $\delta_u$ is a member of the set $\Delta_i$ minus a false positive probability; $i$ is the lesser value among those returned by the set of hash functions.

## 2.2 Homomorphic Encryption

In the protocol proposed in this paper, we use homomorphic encryption. In particular, in the discussion of the protocol we focus on the Paillier cryptosystem [14], an asymmetric encryption scheme that features notable homomorphic properties, although any equivalent cipher could be used in its stead. In general, an encryption scheme has homomorphic properties when it is possible to compute certain operations on a ciphertext without decrypting it and, therefore, without knowledge of the decryption key. In particular, we say an encryption scheme is *additively homomorphic* when an operation on a ciphertext and a plaintext results in the sum of the two plaintexts. We have instead *multiplicative homomorphism* between an encrypted plaintext and a plaintext when an operation results into the multiplication of the two plaintexts. If we identify such operation with the symbol $\boxdot$, the following is true for a multiplicatively homomorphic cipher:

$$\mathrm{Dec}\left(\mathrm{Enc}\left(p_1\right) \boxdot p_2\right) = p_1 \cdot p_2 \ . \tag{6}$$

The Paillier cryptosystem is both additively and multiplicatively homomorphic. In this case, the product of two ciphertexts will decrypt to the sum of their corresponding plaintexts (additive property), while an encrypted plaintext raised to the power of another plaintext will decrypt to the product of the two plaintexts (multiplicative property). Therefore, for the Paillier cipher:

$$\mathrm{Dec}\left(\mathrm{Enc}\left(p_1\right)^{p_2}\right) = p_1 \cdot p_2 \ . \tag{7}$$

This multiplicative property ensures that an encrypted plaintext raised to the power of a constant $k$ will decrypt to the product of the plaintext and $k$.

In the proposed protocol, we apply the multiplicative property to a vector, achieving a secure entrywise product (also known as Hadamard product). We refer to this operation as to *Private Hadamard Product* [2], and we define it in Algorithm 1.

We note here that the Paillier cryptosystem may not be suitable for some heavily computationally constrained devices: however, the proposed protocol can be achieved over any additively homomorphic cipher.

**Algorithm 1:** Private Hadamard product of an encrypted vector of natural numbers for a cleartext binary vector

**Input Alice**: $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$, $\mathbf{X} \in \mathbb{N}^n$.
**Input Bob**: $\mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_n)$, $\mathbf{Y} \in \{0,1\}^n$.
**Output Alice**: $\mathbf{X} \cdot \mathbf{Y}$.

1 Alice generates a public and private key pair using a multiplicative homomorphic encryption scheme.
2 Alice sends to Bob the ciphertext vector $\mathbf{E} = (\mathrm{Enc}(\mathbf{x}_1), \ldots, \mathrm{Enc}(\mathbf{x}_n))$.
3 Bob computes the vector $\mathbf{C} = (\mathrm{Enc}(\mathbf{x}_1) \boxdot \mathbf{y}_1, \ldots, \mathrm{Enc}(\mathbf{x}_n) \boxdot \mathbf{y}_n)$ and sends the result to Alice.
4 Alice uses her secret key to decrypt $\mathbf{C}$ and obtains $\mathbf{D} = \mathrm{Dec}(\mathbf{C}) = (\mathbf{x}_1 \cdot \mathbf{y}_1, \ldots, \mathbf{x}_n \cdot \mathbf{y}_n) = \mathbf{X} \cdot \mathbf{Y}$.

## 3 A Secure Routing Strategy

We study a setting where different, heterogeneous subnetworks are interconnected, creating a larger network. The subnetworks are connected to each other by the *routing layer*, that is, the part of the overarching network infrastructure that manages and routes inter-network communication. Each subnetwork is composed of multiple nodes, and can be connected to the routing layer either directly, or through one or more gateways. In the case of Wireless Sensor Networks, these gateways could also represent the base stations (where information from the sensor node is collected). The aim of our construction is to enable private routing between the subnetworks. In particular, we want to prevent an attacker that controls one or more nodes of the network from being able to learn the topology and structure of the network. Specifically, he should not be able to: determine the number of subnetworks, other than those where he controls a node; the location of any node in the network, that is, to which subnetwork a node belongs. We define the security of our construction as follows:

**Security Definition.** Private routing between different subnetworks in a wider network is achieved when: any node in the network only needs the ID's of other nodes in order to communicate with them, and learns nothing about their position within the network; for each packet received, the routing layer learns only the subnetwork to which the packet should be routed, and nothing about the identity of the sending and receiving nodes. Any subnetwork gateway only routes packets transparently between the subnetwork and the

routing layer, and, similarly to other nodes in the subnetwork, learns nothing about the positions of nodes outside its subnetwork.

The security of the construction is analysed in Section 4.

### 3.1 Routing Strategy

Each node of the network is identified by a unique, random ID. Contrary to the IP address, the ID does not contain or imply any information regarding the network structure. Within the network, nodes communicate using their respective IDs, following a tunneling and encapsulation strategy for lower level protocols (such as TCP/IP) similar to the one used in other private-preserving protocols, including onion routing [7]. In practice, communication between nodes of the network is first tunnelled to the local gateway, then from the gateway to the routing layer, from then to the destination gateway and finally to the destination node. Gateways do not have an active role, and they only relay communication between the nodes in their subnetwork and the routing layer transparently. In general, each party in the communication will not reveal unnecessary information to the following one. The gateway of the sending node, in particular, will not communicate the ID of the node to the routing layer. As the receiving gateway does not know to which node in its subnetwork the communication is destined to, it broadcasts the packets to all nodes in the receiving subnetwork. Since communication is encrypted (as explained in the following), only the intended receiver will be able to decrypt the information. An example of network structure is presented in Figure 2.

### 3.2 Packets and Routing Information

Messages transmitted through the network using the anonymous routing protocol are composed of two parts: a *header*, which contains routing information; and a *payload*, which is encrypted and encapsulates the communication being anonymously routed (in practice, the payload contain packets of lower layer protocols such as UDP or TCP).

In order to encrypt the payload, we assume that each node in the network has a public/private key pair, and a key distribution mechanism exists between the nodes, so that each node knowing another node's ID either knows or can retrieve the node's public key as well (discussed below). Encryption of the payload is performed by the sending node $s$ using the public key $Pk_r$ of the receiving node $r$, which can then decrypt the transmission using its secret key $Sk_r$. As communication is routed anonymously, the ID of the sender is

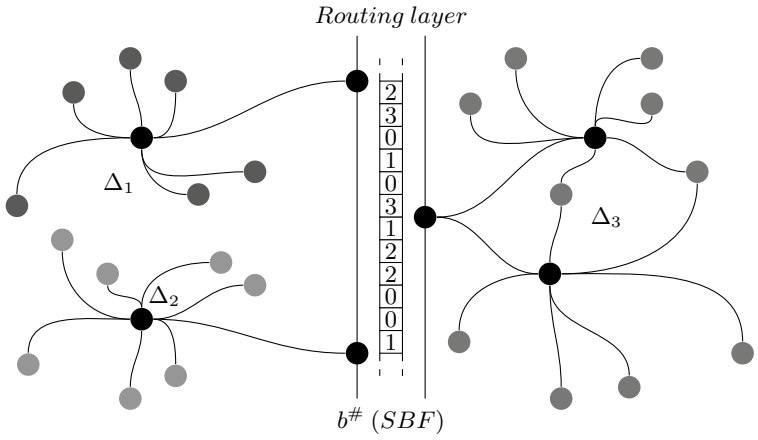


**Figure 2** A sample sensor network composed of three subnetworks $\Delta_1$, $\Delta_2$ and $\Delta_3$. Each subnetwork, composed by a set of nodes, represents an *Area of Interest* (AOI) as described in [2, 16], and is marked with a label. Anonymous routing of packets between the subnetworks (done by the routing layer) is achieved using an SBF representing the network.

included in the encrypted payload as well, in order for the receiving node to be able to respond.

The use of random IDs to identify the nodes removes the need to know the destination IP address in order to initiate communication, and hides the originating IP. It also means that no communication is possible without knowledge of the ID of the destination node. However, in order for the routing to be anonymous, the header does not include the ID of the sending and receiving nodes, but only routing information in the form of an homomorphically encrypted SBF. In particular, the network maintainer builds an SBF representing all the nodes in the network and their respective subnetwork. As shown in Figure 1, the elements of the set over which the SBF is built are the IDs of the nodes, while the sets are the subnetworks, each represented by a label. The SBF built this way, $b^{\#}$, is encrypted using a homomorphic encryption scheme, as explained in Section 2.1. In this construction we use the Paillier cryptosystem [14], but any cipher with equivalent homomorphic properties can be used. In particular, other more lightweight cryptosystems could be more suitable for resource-constrained devices. The secret key $Sk^{\#}$ of the homomorphic key pair is known by the routing layer, while the public key $Pk^{\#}$ and the encrypted filter $\mathrm{Enc}_{Pk^{\#}}\left(b^{\#}\right)$ are distributed to all the nodes. The nodes also know the set of hash functions used in constructing the filter.

**Table 2** Information available to each stakeholder. The first row identifies cryptographic keys owned by the stakeholder and information related to the filter; the second row routing information and IDs of the nodes in the network

| **Node j** | | **Routing Layer** | | **Network Mainteiner** | | | |
|---|---|---|---|---|---|---|---|
| $Enc_{Pk^{\#}}(b^{\#})$<br>Hash set<br>$Pk_j, Sk_j$ | | $Sk^{\#}$ | | $b^{\#}$<br>$Pk^{\#}, Sk^{\#}$<br>Hash set | | (homomorphic key pair) | |
| *Node ID* | *Public key* | *Subnetwork IP* | *Area* | *Node IP* | *Node ID* | *Area* | *Key pair* |
| $ID_1$ | $Pk_1$ | $122.200.64/24$ | 1 | $IP_1$ | $ID_1$ | 1 | $Pk_1, Sk_1$ |
| ... | ... | ... | ... | ... | ... | ... | ... |
| $ID_i$ | $Pk_i$ | $122.200.43/24$ | $k$ | $IP_i$ | $ID_i$ | $k$ | $Pk_i, Sk_i$ |

Table 2 summarizes the information that each party in the protocol needs in order to communicate. The information is divided in two sets: information related to the encryption mechanism (such as public keys), in the upper row; and information related to network communication (including IDs and IP addresses), in the lower row.

In this paper, we assume that knowledge of the ID of a node equates to knowledge of its public key: any suitable key distribution scheme can be applied to achieve this. The key distribution can in fact coincide with the strategy used to notify nodes of new IDs they can communicate with, as the information (both the ID and the public key) needs to be transmitted in order to enable the node to communicate. While a key distribution strategy would be out of the scope of this work, we note that existing schemes designed for anonymity protocols for distributed settings such as [18] can be directly applied to the proposed scenario.

### 3.3 Routing Protocol

The anonymous routing protocol is defined in Algorithm 2. Communication happens between a sender node $s$ and a receiver node $r$ in two different subnetworks ($\Delta_s$ and $\Delta_i$ respectively). In essence, the sender produces a filter containing the receiver's identifier $ID_r$ and performs the private Hadamard product with the encrypted filter $\mathrm{Enc}_{Pk^{\#}}\left(b^{\#}\right)$, obtaining $e^{\#}$. The routing layer can then decrypt the information contained in $e^{\#}$ and learn the destination subnetwork $\Delta_i$. The protocol makes it possible for $s$ to communicate with $r$ without knowing it location within the network. At the same time, the routing layer can route packets without learning their content, nor the sender and receiver identity. The protocol is schematized in Figure 3.

**Algorithm 2:** Anonymous inter-network routing protocol

Communication happens as follows:

1. The sender node $s$ identifies the anonymous identifier $ID_r$ of the receiving node $r$. $s$ then builds an SBF with $ID_r$ as only element, using the known set of hash functions. Once the filter is built, $s$ counts the number $z$ of non-zero values in it. Then, the node performs the private Hadamard product between the filter it just built and the encrypted filter $\mathrm{Enc}_{Pk^{\#}}\left(b^{\#}\right)$, using the multiplicative homomorphic properties of the cryptosystem. We call the resulting combined encrypted filter $e^{\#}$. Finally, the sender shuffles $e^{\#}$, and sends it to the gateway, with $z$ and the encrypted payload $\mathrm{Enc}_{Pk_r}\left(msg\right)$.
2. The sending gateway relays transparently the information received by $s$ to the routing layer.
3. The routing layer decrypts $e^{\#}$: the decrypted filter is composed of zeros, and a number of non-zero values. If the number of non-zero values is equal to $z$, then the receiving node $r$ exists. The smallest value $i$ among the non-zero ones identifies the correct subnetwork to which the communication will be routed (see Section 2). Finally the routing layer transmits the encrypted payload $\mathrm{Enc}_{Pk_r}\left(msg\right)$ to the correct subnetwork $\Delta_i$.
4. The gateway of $\Delta_i$ receives the encrypted payload and broadcasts it to all the nodes in the subnetwork.
5. The intended receiver $r$ receives $\mathrm{Enc}_{Pk_r}\left(msg\right)$ and decrypts it using its secret key $Sk_r$.

The properties of the spatial Bloom filters introduce the possibility of false positives and inter-set errors: in the first scenario, an element outside the sets over which the filter has been built could be recognized as member of a set; in the inter-set error scenario, an element that is a member of a set $X$ could be recognized as member of set $Y$. The former case has no real implications for the proposed protocol: it would only apply to the case of a node in the network using non-existing or unknown IDs. But as no public key is associated to these IDs, communication is impossible. The latter case could result in the wrong routing being applied to the communication: however, we note that the probability of this event can be calibrated through the use of appropriate parameters (such as the length of the filter and the number of hash functions) during the filter construction, and a filter can be tested after it has been built (testing for membership all the elements of the construction set $\bar{S}$) to verify that no inter-set errors are possible.

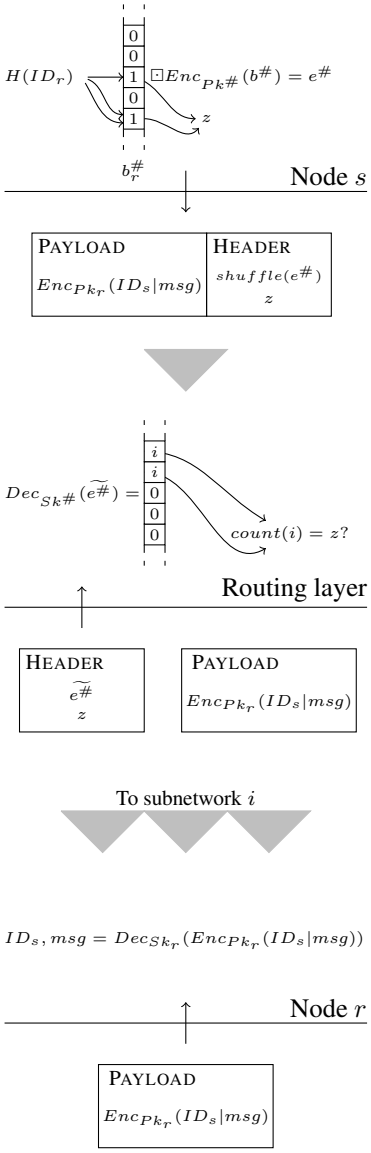


**Figure 3** Operation of the private routing protocol. Node $s$ wants to securely transmit message $msg$ to node $r$. Node $r$ belongs to subnetwork $i$, but $s$ only knows $r$'s ID ($ID_r$). Communication proceeds as follows: $s$ generates the SBF related to $ID_r$ and counts the number $z$ of non-zero values in it; the filter is then multiplied (through an homomorphic encryption operation) by the shared encrypted filter $b^{\#}$. The resulting filter is then sent to the routing layer, together with $z$. The routing layer decrypts it, and computes the destination subnetwork $i$. The payload (that is, the encrypted message) is then routed to the subnetwork $i$ and node $r$, either through a gateway or by broadcast. $r$ receives the message and decrypts it.

## 4 Security Analysis

In order to analyse the security of our construction, we discuss three separate scenarios: in the first, an attacker gains control over a node in the network; in the second, the attacker controls a subnetwork gateway, and in the third, the attacker controls the routing layer (or part of it). In all three cases, we assume the attacker will not actively disrupt network traffic, but will limit himself to observing traffic visible to him in order to learn information on the network structure and topology (context information). This is called a *semi-honest behaviour*. In the following, we show how in each of the three cases the attacker is unable to learn any meaningful information on the network structure, and therefore the security definition is satisfied. Security cannot be guaranteed in case the attacker controls simultaneously 1) the routing layer and 2) either one or more nodes, or one or more gateways, or a combination of the two. The extent to which security is compromised in this case depends on the number of nodes and gateways controlled, and is limited to the parts of the network the attacker has visibility of.

**Attacker controlling a node.** In this case, the attacker can read all information sent and received by the node, and learns the IDs of all the other nodes with which the controlled node can communicate. The attacker also learns the encrypted filter, but has no information to decrypt it. The attacker cannot learn the IP addresses corresponding to the nodes, as they are unknown to the controlled node and cannot be derived from the respective IDs. Similarly, the attacker cannot learn the network structure (the position of the nodes within the subnetworks and the number of subnetworks), as the routing of sent and received packets is achieved anonymously.

**Attacker controlling a subnetwork gateway.** An attacker controlling a gateway will learn all the identity of all the nodes in the respective subnetwork. However, he will not be able to read any information sent and received by the nodes, as the payloads are encrypted. Similarly, he will not learn the destination of sent packets or the origin of received ones, as the routing information $e^{\#}$ is encrypted. Finally, the attacker cannot learn the network structure as per the case above.

**Attacker controlling the routing layer.** In this case, the attacker will be able to watch the flow of information between the different subnetworks. However, due to the properties of the SBF, even being able to decrypt the encrypted routing information $e^{\#}$ will not enable him to learn the identity of

the receiving node $r$. Similarly, he cannot learn the identity of the sending node $s$, as this is encrypted within the payload, and the sending gateway will not communicate it to him.

## 5 Conclusions

In this paper, we present a private routing protocol that can be used to communicate anonymously between different networks. Our protocol can be applied in a variety of Internet of Things scenarios: from Wireless Sensor Networks, to interconnected IoT systems composed by different devices or infrastructures.

Our protocol achieves context privacy by using homomorphic encryption, tunnelling and the spatial Bloom filters. In particular, we achieve the following properties: communication between nodes can only be read by the intended receiver; the network structure and topology (context information) is kept private to all nodes; the identity and location of the sending and receiving nodes in two different subnetworks is kept private to the routing layer; and the routing layer is oblivious to the origin and destination of any communication between subnetworks. These properties enable context privacy and security against adversaries who control one or more nodes within the network, or even the routing layer. Therefore, the proposed anonymous routing protocol can prevent attacks aimed at taking over control of the network.

## References

[1] Burton H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM*, 13(7):422–426, 1970.

[2] Luca Calderoni, Paolo Palmieri, and Dario Maio. Location privacy without mutual trust: The spatial bloom filter. *Computer Communications*, 68:4–16, 2015. Security and Privacy in Unified Communications: Challenges and Solutions.

[3] Shaibal Chakrabarty, Monica John, and Daniel W. Engels. Black routing and node obscuring in iot. In *3rd IEEE World Forum on Internet of Things, WF-IoT 2016, Reston, VA, USA, December 12–14, 2016*, pages 323–328. IEEE Computer Society, 2016.

[4] Mauro Conti, Jeroen Willemsen, and Bruno Crispo. Providing source location privacy in wireless sensor networks: A survey. *IEEE Communications Surveys and Tutorials*, 15(3):1238–1280, 2013.

[5] Jing Deng, Richard Han, and Shivakant Mishra. Intrusion tolerance and anti-traffic analysis strategies for wireless sensor networks. In *2004*

*International Conference on Dependable Systems and Networks (DSN 2004), Proceedings*, page 637. IEEE Computer Society, 2004.

[6] Jing Deng, Richard Han, and Shivakant Mishra. Decorrelating wireless sensor network traffic to inhibit traffic analysis attacks. *Pervasive and Mobile Computing*, 2(2):159–186, 2006.

[7] Roger Dingledine, Nick Mathewson, and Paul F. Syverson. Tor: The second-generation onion router. In Matt Blaze, editor, *Proceedings of the 13th USENIX Security Symposium*, pages 303–320. USENIX, 2004.

[8] Santiago Gaitan, Luca Calderoni, Paolo Palmieri, Marie-Claire Ten Veldhuis, Dario Maio, and M. Birna Van Riemsdijk. From sensing to action: Quick and reliable access to information in cities vulnerable to heavy rain. *IEEE Sensors Journal*, 14(12):4175–4184, 2014.

[9] Shahabeddin Geravand and Mahmood Ahmadi. Bloom filter applications in network security: A state-of-the-art survey. *Computer Networks*, 57(18):4047–4064, 2013.

[10] Pandurang Kamat, Yanyong Zhang, Wade Trappe, and Celal Ozturk. Enhancing source-location privacy in sensor network routing. In *25th International Conference on Distributed Computing Systems (ICDCS 2005)*, pages 599–608. IEEE Computer Society, 2005.

[11] Na Li, Nan Zhang, Sajal K. Das, and Bhavani M. Thuraisingham. Privacy preservation in wireless sensor networks: A state-of-the-art survey. *Ad Hoc Networks*, 7(8):1501–1514, 2009.

[12] Shancang Li and Li Xu. *Securing the Internet of Things*. Elsevier, January 2017.

[13] Yingshu Li, My T. Thai, and Weili Wu, editors. *Wireless Sensor Networks and Applications*. Signals and Communication Technology. Springer, 2008.

[14] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In Jacques Stern, editor, *Advances in Cryptology – EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2–6, 1999, Proceeding*, volume 1592 of *Lecture Notes in Computer Science*, pages 223–238. Springer, 1999.

[15] Paolo Palmieri. Preserving context privacy in distributed hash table wireless sensor networks. In Sihan Qing, Eiji Okamoto, Kwangjo Kim, and Dongmei Liu, editors, *Information and Communications Security – 17th International Conference, ICICS 2015, Beijing, China, December 9–11, 2015, Revised Selected Papers*, volume 9543 of *Lecture Notes in Computer Science*, pages 436–444. Springer, 2015.

[16] Paolo Palmieri, Luca Calderoni, and Dario Maio. Spatial bloom filters: Enabling privacy in location-aware applications. In Dongdai Lin, Moti Yung, and Jianying Zhou, editors, *Information Security and Cryptology – 10th International Conference, Inscrypt 2014, Beijing, China, December 13–15, 2014, Revised Selected Papers*, volume 8957 of *Lecture Notes in Computer Science*, pages 16–36. Springer, 2014.

[17] Paolo Palmieri, Luca Calderoni, and Dario Maio. Private inter-network routing for wireless sensor networks and the internet of things. In *Proceedings of the ACM International Conference on Computing Frontiers, CF'17, Siena, Italy, May 15–18, 2017*, 2017. To appear.

[18] Paolo Palmieri and Johan A. Pouwelse. Key management for onion routing in a true peer to peer setting. In Maki Yoshida and Koichi Mouri, editors, *Advances in Information and Computer Security – 9th International Workshop on Security, IWSEC 2014. Proceedings*, volume 8639 of *Lecture Notes in Computer Science*, pages 62–71. Springer, 2014.

[19] Michael G. Solomon, Vaidy S. Sunderam, Li Xiong, and Ming Li. Enabling mutually private location proximity services in smart cities: A comparative assessment. In *IEEE International Smart Cities Conference, ISC2 2016, Trento, Italy, September 12–15, 2016*, pages 1–8. IEEE, 2016.

[20] Yong Xi, Loren Schwiebert, and Weisong Shi. Preserving source location privacy in monitoring-based wireless sensor networks. In *20th International Parallel and Distributed Processing Symposium (IPDPS 2006), Proceedings*. IEEE, 2006.

[21] Yi Yang, Min Shao, Sencun Zhu, Bhuvan Urgaonkar, and Guohong Cao. Towards event source unobservability with minimum network traffic in sensor networks. In Virgil D. Gligor, Jean-Pierre Hubaux, and Radha Poovendran, editors, *Proceedings of the First ACM Conference on Wireless Network Security, WISEC 2008*, pages 77–88. ACM, 2008.

[22] Liang Zhang. A self-adjusting directed random walk approach for enhancing source-location privacy in sensor network routing. In Seizo Onoe, Mohsen Guizani, Hsiao-Hwa Chen, and Mamoru Sawahashi, editors, *Proceedings of the International Conference on Wireless Communications and Mobile Computing, IWCMC 2006*, pages 33–38. ACM, 2006.

## Biographies

**Paolo Palmieri** is a Lecturer in Cyber Security at the Centre for Electronic Warfare Information and Cyber, Cranfield University (UK). He holds a Ph.D. in cryptography from the Université Catholique de Louvain (Belgium). His research work focuses on cryptographic protocols for privacy and anonymity, and he has worked on privacy enhancing technologies, secure computation, location privacy, and the security of smart cities and the Internet of Things.

**Luca Calderoni** received a Ph.D. degree in computer science from the University of Bologna, Italy, in 2015. He is currently a Post-doctoral Researcher with the Smart City Laboratory of the University of Bologna, in Cesena, Italy. His research activity focuses on privacy and security in digital systems and smart cities. He has published on location privacy, border controls, secure and privacy-preserving tracking and monitoring technologies, location-aware applications and urban ICT infrastructures.

**Dario Maio** received a Master's degree in electronic engineering from the University of Bologna, Italy in 1975. He is a Full Professor of Information Systems with the Department of Computer Science and Engineering, University of Bologna. He is a member of IEEE, ACM and IAPR. He was the Chair of the Cesena Campus (2001–2007), and is the Director of the BioLab and the Coordinator of the Smart City Lab with the University of Bologna. He has published more than 200 research papers investigating various aspects of computer science including distributed computer systems, computer performance evaluation, database design, information systems, neural networks, autonomous agents, pattern recognition, and biometric systems.